Microsoft PLR 4-2, Exhibit B: Mini Markman Preliminary Claim Construction – Claim Terms[1]

Claim Term	Preliminary Construction
access, accessed, access to, accessing	Establishing the connections, routings, and **security** requisites needed to physically obtain something. Access to protected information is required, but insufficient, for **use** of that information. In VDE, access to protected information is achieved only through execution (within a **Secure Processing Environment**) of the **VDE control**(s) assigned to the particular "access" request, satisfaction of all requirements imposed by such execution, and the **controlled opening** of the **secure container containing** the information.
addressing	Referring by specific location or individual name to something without physically storing it.
allowing, allows	Actively permitting an action that otherwise cannot be taken (i.e., is absolutely prohibited) by any user, process, or device. In VDE, an action is allowed only through execution (within a **Secure Processing Environment**) of the **VDE control**(s) assigned to the particular action request, and satisfaction of all requirements imposed by such execution.
applying . . . in combination	[This shall be construed in connection with a disputed claim phrase.]
arrangement	[This shall be construed in connection with a disputed claim phrase.]
aspect	[This shall be construed in connection with a disputed claim phrase.]
associated with	1. A specific, direct, persistent, and binding relationship with one or more discrete items. Code that processes information but is merely a general-purpose component of an installation is not "associated with" that information. In VDE, an association between a unit of **executable** code and particular information, or between particular **control information** and a **secure container**, cannot be broken except as **allowed** by execution (within a **Secure Processing Environment**) of assigned **VDE control(s)** and satisfaction of all requirements imposed by such execution. 2. Associations in **VDE** are created with a **component assembly**, a **secure container**, a **Secure Processing Environment**, "object registration," and other mechanisms of **VDE** for (allegedly) individually ensuring the "access control" "handcuffs" between specific **controls**, specific objects (and their content at an arbitrary granular level), and specific users.

[1] The word "invention" is used not to suggest that anything described in InterTrust's patents in fact was novel or non-obvious or inventive, but rather to identify what was described as the **alleged** invention. Also, features and capabilities are described as they are described in the InterTrust patent application, even though the patent application did not describe an actual working system having any of these capabilities. Also, Microsoft's proposed constructions use many terms from the InterTrust patents that are used inconsistently or otherwise indefinitely in the patents. Those terms are used by Microsoft in their narrowest applicable sense, and without waiving the right to assert the indefiniteness of this claim language. Also, the preliminary constructions assume (without conceding) that the February, 1995, InterTrust patent application was incorporated by reference into the '721, '861, and '683 patents, effectively for claim construction purposes. If the Court concludes otherwise, then the proper constructions will be different in some cases. Bolded terms are preliminarily defined in Exhibits A-C of Microsoft's PLR 4-2 papers.

Claim Term	Preliminary Construction
authentication	The act of verifying credentials designed to vouch for the **authenticity** of the identity, data **integrity**, and origin **integrity** of a person, device, program, information, or process.
authorization information, authorized, not authorized	authorized: An action is permitted that otherwise cannot be taken (i.e., is absolutely prohibited) by any user, process, or device. In VDE, an action is authorized only through execution of the applicable **VDE control**(s) within a **VDE Secure Processing Environment** and satisfaction of all requirements imposed by such execution. authorization information: **"Control information"** identifying the exact modular code components to be assembled into a **VDE control** and executed within a **Secure Processing Environment** to permit a particular activity that otherwise cannot be taken (i.e., is absolutely prohibited). ("Control information" is information which identifies the exact modular code components and data which must be assembled and executed to **control** a particular activity on particular information, of arbitrary, user-defined granularity, by particular user(s)). "not authorized": The action is prohibited and cannot be taken by any user, process, or device.
budget control; budget	budget: A unique type of **"method"** that specifies limitations on future usage (e.g., copying) of digital information and how such usage will be paid for, if at all. (A "method" is a collection of basic instructions, and information related to basic instructions, that provides context, data, requirements, and/or relationships for use in performing, and/or preparing to perform, basic instructions in relation to the operation of one or more electronic appliances.) budget control: A **VDE control** assembled using a budget, and enforcing that budget. No process, user, or device is able to make the use identified by the budget once the budget's specified limitation on that **use** has been met.
can be	Something is permitted that otherwise cannot happen (i.e., is absolutely prohibited).
capacity	Available storage space that is still capable of allocation. For example, a 650 MB blank CD, after sealing, has zero capacity because no new material may be stored within it.
clearinghouse	A computer system that provides intermediate storing and forwarding services for both content and **audit information**, and which two or more parties trust to provide its services independently because it is operated under constraint of **VDE security**. "Audit information" means all information created, stored, or reported in connection with an **"auditing"** process. "Auditing" means tracking, metering and reporting the usage of particular information or a particular appliance.
compares, comparison	A processor operation that evaluates two quantities and sets one of three flag conditions as a result of the comparison – greater than, less than, equal to.
component assembly (2)	A cohesive executable component created by a channel which binds or links together two or more independently deliverable **load modules**, and associated data. A component assembly is assembled, and executes only within a **VDE Secure Processing Environment**. A component assembly is assembled dynamically in response to, and to service, a particular content-related activity (e.g., **use** request). Each VDE component assembly is assigned and dedicated to a particular activity, particular user(s), and particular protected information. Each component assembly is independently

Claim Term	Preliminary Construction
	assembled, loadable and deliverable vis-à-vis other component assemblies. The dynamic assembly of a component assembly is directed by a "blueprint" record **containing control information for** this particular activity on this particular information by this particular user(s). Component assemblies are extensible and can be configured and reconfigured (modified) by all users, and combined by all users with other component assemblies, subject only to other users' "senior" **controls**.
contain, contained, containing	Physically storing within, as opposed to **addressing**.
control (n.), controls (n.) (2 – 193:1,11,15,19; 891:1)	VDE **allows access** to or **use** of protected information only through execution of (and satisfaction of the requirements imposed by) independent, special-purpose, executable VDE control(s). A VDE control can execute only within a **Secure Processing Environment**. Each VDE control is a **component assembly** dedicated to a particular activity (e.g., editing, modifying another control, a user-defined action, etc.), particular user(s), and particular protected information. Each separate information **access** or **use** is independently **controlled** by independent VDE control(s). Each VDE control is assembled within a **Secure Processing Environment** from independently deliverable modular components (e.g., **load modules** or other controls), dynamically in response to an information **access** or use **request**. The dynamic assembly of a control is directed by a "blueprint" **record** (put in place by one or more VDE users) **containing control information** identifying the exact modular code components to be assembled and executed to **govern** this particular activity on this particular information by this particular user(s). Each control is independently assembled, loaded and delivered vis-à-vis other controls. **Control information** and controls are extensible and can be configured and modified by all users, and combined by all users with any other VDE **control information** or controls (including that provided by other users), subject only to "senior" user controls. Users can assign **control information** (including alternative **control information**) and **controls** to an arbitrarily fine, user-defined portion of the protected information , such as a single paragraph of a document, as opposed to being limited to file-based controls. VDE controls reliably limit use of the protected information to **authorized** activities and amounts.
controlling, control (v.)	1. Reliably defining and enforcing the conditions and requirements under which an action that otherwise absolutely cannot be taken, will be **allowed**, and the manner in which it may occur. Absent verified satisfaction of those conditions and requirements, the action cannot be taken by any user, process or device. In VDE, an action is controlled through execution of the applicable **VDE control**(s) within a **VDE Secure Processing Environment**. 2. More specifically, in VDE, **controlling** is effected by use of **VDE controls**, **VDE secure containers**, and VDE foundation (including **VDE Secure Processing Environment**, "object registration," and other mechanisms for allegedly individually ensuring that specific **controls** are enforced vis-à-vis specific objects (and their content at an arbitrary granular level) and specific "users.")
copied file	A **digital file** which has been **copied** at least once, not the **copy** itself. A "copy" is what is formed by a **copying** operation, and it may or may not be encrypted, ephemeral, usable, or accessible.
copy, copied, copying (v.)	To duplicate a **digital file** or other complete physical block of data from one location on a storage medium to another location on the same or different storage medium, leaving the original block of data unchanged, such that two distinct and independent objects exist. Although the layout of the data values in physical storage may differ from the original, the resulting "copy" is logically indistinguishable from the original. The resulting "copy" may or

Claim Term	Preliminary Construction
	may not be encrypted, ephemeral, usable, or accessible.
copy control	A **VDE control** which **controls** some **access** to or **use** of a **copy**.
creating, creation	[This shall be construed in connection with a disputed claim phrase.]
data item	An individual unit of information representing a single value, such as that stored in a field of a larger **record** in a database. It is the smallest useful unit of named information in the system.
derive, derives	To retrieve from a specified source.
descriptive data structure	A machine-readable data structure (e.g., text file, template, **secure container**, etc.) **containing** or **addressing** descriptive information (e.g., **metadata**, shorthand abstract representation, integrity constraints, **rules**, instructions, etc.) about (1) the layout, generic format (e.g., location of a particular type of information), attributes, or hierarchical structure (e.g., file hierarchy) of the contents section of one or a family of other data structure(s) (e.g., **secure container**, other rights management related structure, etc.), (2) the operations or processes used to create or **use** such other data structure(s) (e.g., **rules** for handling the data structure), and/or (3) the consequences of such operations (e.g., billing the user a certain fee for printing). The descriptive data structure is capable of being used to create or handle (e.g., read, locate information within, request information from, and/or manipulate) the other data structure(s). The descriptive data structure is not **associated with** the other data structure(s) and does not **contain** or specify its particular contents (e.g., "Yankees Win the Pennant!").
designating	[This shall be construed in connection with a disputed claim phrase.]
device class	The generic name for a group of device types. For example, all display stations belong to the same device class. A device class is different from a device type. A device type is composed of all devices that share a common model number or family (e.g. IBM 4331 printers).
digital file	A static unit of storage allocated by a **"file system"** and **containing** digital information. A digital file enables any application using the **"file system"** to randomly access its contents and to distinguish it by name from every other such unit. A **copy** of a digital file is a separate digital file. (A "file system" is the portion of the operating system that translates requests made by application programs for operations on "files" into low-level tasks that can control storage devices such as disk drives.)
digital signature, digitally signing	digital signature: An unforgable string of characters (e.g., bits) generated by a cryptographic transformation to a block of data using some secret, which string can be generated only by an agent that knows the secret, and hence provides evidence that the agent must have generated it. digitally signing: Creating a digital signature using a secret **key**. (In symmetric key cryptography, a "secret key" is a **key** that is known only to the sender and recipient. In asymmertric key cryptography, a "secret key" is the private key of a public/private key pair, in which the two keys are related uniquely by a predetermined mathematical relationship such that it is computationally infeasible to determine one from the other.)
entity, entity's control	entity: Any person or organization.

Claim Term	Preliminary Construction
	entity's control: **Control** created, modified, or selected by any person or organization to **control** a particular **use** of or **access** to particular protected information by a particular user(s).
environment	[This will be construed in connection with other disputed claim terms.]
executable programming, executable (2)	executable: A cohesive series of machine code instructions in a format that can be loaded into memory and run (executed) by a connected processor. executable programming: A cohesive series of machine code instructions, comprising a **computer program**, in a format that can be loaded into memory and run (executed) by a connected processor. (A "computer program" is a complete series of definitions and instructions that when executed on a computer will perform a required or requested task.)
execution space, execution space identifier	execution space: A processor-addressable physical **memory** into which data and executable code can be loaded, which is assigned to a single executing process while that process is actively executing. **Memory** holding "swapped out" processes or **executables** is not part of an "execution space." execution space identifier: A value that uniquely **identifies** a particular execution space.
generating	[This shall be construed in connection with a disputed claim phrase.]
govern, governed, governed item, governing	govern, governing, governed: See **control** (v.) governed item: Information, of arbitrarily fine granularity, whose access and use by any user, process, or device which is **controlled**.
halting	Stopping execution of a running (executing) process unconditionally (i.e., without providing any specific condition for resumption). For example, executing an instruction known as a "breakpoint halt instruction."
host processing environment	A **processing environment** within a **VDE** node which is not a **Secure Processing Environment**. A "host processing environment" may either be "secure" or "not secure." A "secure" host processing environment is a self-contained **protected processing environment**, formed by loaded, **executable programming** executing on a general purpose CPU (not a **Secure Processing Unit**) running in protected (privileged) mode. A "non-secure" host processing environment is formed by loaded, **executable programming** executing on a general purpose CPU (not a **Secure Processing Unit**) running in user mode.
identifier, identify, identifying	identifier: Any text string used as a label naming an individual instance of what it identifies. identify: To establish as being a particular instance of a person or thing.
including	(With respect to a **digital file, control, authorization information, Secure Processing Environment, descriptive data structure**, element, **load module**, header, or **secure container**): Physically storing within, as opposed to **addressing**.
information previously stored	Information that once was stored but is no longer stored.

Claim Term	Preliminary Construction
integrity programming	**Executable programming** that when executed checks and reports on the **integrity** of a device or process. "Integrity" means the property that information has not been altered either intentionally or accidentally.
key	A bit sequence used and needed by a cryptographic algorithm to encrypt a block of plain text or to decrypt a block of cipher text. A key is different from a key seed or other information from which the actual encryption and/or decryption key is constructed, **derived**, or otherwise identified. In symmetric key cryptography, the same key is used for both encryption and decryption. In asymmetric or "public key" cryptography, two related keys are used; a block of text encrypted by one of the two keys (e.g., the "public key") can be decrypted only by the corresponding key (e.g., the "private key").
load module (2)	An executable, modular unit of machine code suitable for loading into **memory** for execution by a processor. A load module is encrypted (when not within a **Secure Processing Unit**) and has an **identifier** that a calling process must provide to be able to use the load module. A load module is combinable with other load modules, and associated data, to form executable **component assemblies**. A load module can execute only in a **VDE protected processing environment**.
machine check programming	**Executable programming** that when executed generates a unique "machine signature" which distinguishes the physical machine from all other machines. This machine check programming code sometimes is invoked by **integrity programming**.
metadata information (2 (metadata))	Data that describes other data managed within an application or environment, such as its meaning, representation in storage, what it is used for and by whom, context, quality and condition, or other characteristics. Metadata may describe data elements or attributes (name, size, data type, etc) and data about **records** or data structures (length, fields, columns, etc) and data about data (where it is located, how it is associated, ownership, etc.).
opening secure containers	Establishing the requisites needed to attempt to **access** the contents of a **secure container**. Opening is a necessary but insufficient step before the contents of a **secure container** may be copied, decrypted, read, manipulated, or otherwise **used**, or **accessed**. No process, user, or device may **access** or **use** the contents of a **secure container** without first opening that **secure container**. A **secure container** may be opened only through execution of the assigned **VDE control**(s) within a **VDE Secure Processing Environment** and satisfaction of all requirements imposed by such execution.
operating environment	See **processing environment**.
organization, organization information, organize	organization, organization information: The manner in which data is represented and laid out in physical storage. For example, for data organized as records: the field hierarchy, order, type and size. organize: Representing and laying out data in a particular manner in physical storage.
portion	[This shall be construed in connection with a disputed claim phrase.]
prevents	Imposes an active restraint on an action such that it absolutely cannot occur by any means or under any circumstances.
processing environment (2 – 912:35, 900:155, 721:34)	A standardized, well-defined, self-contained, computing base, formed by hardware and executing code, that provides an "interface" and set of resources which can support different applications, on different types of hardware platforms. In the context of claim 35 of the '912 patent: a **Secure Processing**

Claim Term	Preliminary Construction
	Environment.
protected processing environment (2 – 721:34)	1. A uniquely identifiable, self-contained computing base trusted by all VDE nodes to protect the **availability, secrecy, integrity** and **authenticity** of all information identified in the patent application as being protected, and to guarantee that such information will be **accessed** and **used** only as expressly **authorized** by **VDE controls**. At most VDE nodes, the protected processing environment is a **Secure Processing Environment** which is formed by, and requires, a hardware **tamper-resistant barrier encapsulating** a special-purpose **Secure Processing Unit** having a processor and internal **secure memory**. ("Encapsulated" means hidden within an object so that it is not directly accessible but rather is accessible only through the object's restrictive interface.) The **barrier prevents** all **unauthorized** (intentional or accidental) interference, removal, observation, and **use** of the information and processes within it, by all parties (including all users of the device in which the Protected Processing Environment resides), except as expressly **authorized** by **VDE controls**. A Protected Processing Environment is under **control** of **controls** and **control information** provided by one or more parties, rather than being under **control** of the appliance's users or programs. Where a VDE node is an established financial clearinghouse, or other such facility employing physical facility and user-identity authentication security procedures trusted by all VDE nodes, and the VDE node does not **access** or **use** VDE-protected information, or assign VDE **control information**, then the Protected Processing Environment at that VDE node may instead be formed by a general-purpose CPU that executes all VDE "security" processes in protected (privileged) mode. 2. A Protected Processing Environment requires more than just verifying the **integrity** of digitally signed **executable programming** prior to execution of the programming; or concealment of the program, associated data, and execution of the program code; or use of a password as its protection mechanism.
protecting	Maintaining the **security** of.
record (n.)(2)	A data structure that is a collection of fields (elements), each with its own name and type. Unlike an array, whose elements are accessed using an index, the elements of a record are accessed by name. A record can be accessed as a collective unit of elements, or the elements can be accessed individually.
required	A condition without which an action cannot occur. A required condition acts prospectively – it does not apply to a description created at or after the creation of the object to which it applies.
resource processed	A **record** containing **control information**, which **record** is stored and acted upon within a **processing environment**.
rule (2)	A lexical statement that states a condition under which **access** to or **use** of VDE-protected data will be **allowed** by a **VDE control**. A rule may specify how, when, where, and by whom a particular activity on particular information is to be **allowed**.
secure (2)	A state in which all users of a system are guaranteed that all information, processes, and devices within the system, shall have their **availability, secrecy, integrity** and **authenticity** maintained against all of the identified threats thereto. "Availability" means the property that information is accessible and usable upon demand by authorized persons, at least to the extent that no user may delete the information without **authorization**. "Secrecy," also referred to as confidentiality, means the property that information (including computer processes) is not made available or disclosed to unauthorized persons or processes. "Integrity" means the property that information has not been altered either intentionally or accidentally. "Authenticity" means the

Claim Term	Preliminary Construction
	property that the characteristics asserted about a person, device, program, information, or process are genuine and timely, particularly as to identity, data integrity, and origin integrity.
secure container	A VDE secure container is a self-contained, self-protecting data structure which (a) **encapsulates** information of arbitrary size, type, format, and **organization**, including other, nested, containers, (b) cryptographically protects that information from all **unauthorized access** and **use**, (c) provides encrypted storage management functions for that information, such as hiding the physical storage location(s) of its protected contents, (d) permits the **association** of itself or its contents with **controls** and **control information governing access to** and **use** thereof, and (e) **prevents** such **use** or **access** (as opposed to merely **preventing** decryption) until it is "opened." A secure container can be opened only as expressly **allowed** by the **associated VDE control(s)**, only within a **Secure Processing Environment**, and only through decryption of its encrypted header. A secure container is not directly accessible to any non-VDE or user calling process. All such calls are intercepted by VDE. The creator of a secure container can assign (or allow others to assign) **control information** to any arbitrary portion of a secure container's contents, or to an empty secure container (to **govern** the later addition of contents to the container, and **access** to or **use** of those contents). A container is not a secure container merely because its contents are encrypted and signed. A secure container is itself **secure**. All VDE-protected information (including protected content, information about content usage, and content-**control information, controls**, and **load module**) is **encapsulated** within a secure container whenever stored outside a **Secure Processing Environment** or **secure database**.
secure container governed item	A **governed item protected** by a **secure container**. A secure container governed item may not be **accessed** or **used** in any way, by any user, process, or device, except as **allowed** by its **associated VDE control(s)** executing in a **VDE Secure Processing Environment** and satisfaction of all requirements imposed by such execution.
secure container rule	A **rule protected** by a **secure container**. A secure container rule may not be **accessed** or **used** in any way, by any user, process, or device, except as **allowed** by its **associated VDE control(s)** executing in a **VDE Secure Processing Environment** and satisfaction of all requirements imposed by such execution.
secure database	A data store isolated from all users such that it is **protected** from external observation; and accidental or intentional alteration or destruction. In VDE, a secure database stores tracking, billing, payment, and **auditing** data until the data is delivered **securely** to an authorized **clearinghouse**.
secure execution space	An allocated portion of the **secure memory** within a special-purpose **Secure Processing Unit** which is isolated from the rest of the world, and **protected** from observation by (and **encapsulated** within) a **tamper resistant barrier** and **protected** from alteration by the processor. The processor cryptographically verifies the **integrity** of all code loaded from secure memory prior to execution, executes only the code that the processor has **authenticated** for its use, and is otherwise **secure**.
secure memory, memory	memory: A medium in which data (including executable instructions) may be stored and from which it may be retrieved. "Memory" does not include a "virtual memory." secure memory: A processor-addressable **memory** within a special-purpose **Secure Processing Unit** which is isolated from the rest of the world by (and **encapsulated** within) a **tamper resistant barrier**. "Processor-addressable" means that a connected processor can use the secure memory's

Claim Term	Preliminary Construction
	physical addresses as the operand in a processor instruction such as LOAD or STORE or equivalent instruction. A "memory" is not a "secure memory" merely because it stores encrypted, signed, and/or sealed data; is accessible from a **Protected Processing Environment**; or is within an appliance that is located at a trusted facility with non-VDE physical security and user-identity authentication procedures.
secure operating environment, said operating environment	Same as **Secure Processing Environment**.
securely applying (2 – securely)	securely: Performed in a **Secure Processing Environment** in a manner that guarantees that each affected information or process remains **secure**. securely applying: **securely** (1) executing the applied **executables** (e.g., **controls**) within a **VDE secure execution space**, (2) validating and verifying the **authenticity** and **integrity** of each **executable**, and (3) ensuring that the **executables** are applied only in ways that are intended by the **VDE** participants who created the **executables**.
securely assembling	**securely** (1) linking or binding plural distinct elements together in a particular manner (specified by **authenticated** assembly instructions) into a single cohesive executable unit so the elements can directly reference each other element within the resulting assembly, within a **VDE Secure Processing Environment**, (2) validating and verifying the **authenticity** and **integrity** of each element (e.g., that it has not been modified from or substituted for the correct element) immediately prior to binding it into the assembly, and (3) ensuring that the elements are linked together only in ways that are intended by the **VDE** participants who created the elements and/or specified the assembly thereof.
securely processing	Executing code in a **secure execution space** to act upon some information, in a manner that ensures that the information and the processing remain **secure**.
securely receiving	Receiving digital information in a **secure container**, as part of a communication encrypted on the communications level, at a **Secure Processing Environment authenticated** in accordance with **VDE controls associated with the secure container**.
security (2)	See **secure**.
security level, level of security	An ordered measure of the degree of **security**. The "security level" is persistent unless expressly noted to exist only some of the time. Also, the combination of a hierarchical classification and a set of nonhierarchical categories that represents the sensitivity of an object or the clearance of a subject. For example, Unclassified, Confidential, Secret, and Top Secret are hierarchical classifications, whereas NATO and NOFORN are non-hierarchical categories defined by the DoD Trusted Computing guidelines.
specific information, specified information	[This will be construed in connection with disputed claim phrases]
tamper resistance (2 – tamper)	tamper resistance: The abllity of a **tamper resistant barrier** to **prevent access**, observation, and interference with information or processing **encapsulated** by the barrier.

Claim Term	Preliminary Construction
	tamper: See **tampering.**
tamper resistant barrier	An active device that **encapsulates** and separates a **Protected Processing Environment** from the rest of the world. It prevents information and processes within the **Protected Processing Environment** from being observed, interfered with, and leaving except under appropriate conditions ensuring **security**. It also **controls** external **access** to the **encapsulated secure** resources, processes and information. A **tamper resistant barrier** is capable of destroying **protected** information in response to **tampering** attempts.
tamper resistant software	Software that is **encapsulated** and executed wholly within a **tamper resistant barrier.**
tampering (2)	Attempting to circumvent a **tamper resistant barrier** or other mechanism designed to protect against the observation, **access**, or alteration of data, code, or process execution, or making any unauthorized access, observation, or interference.
use (n.)	Any action with respect to information (e.g., copying, printing, decrypting, executing) other than **access**. In VDE, information use is **allowed** only through execution of the applicable **VDE control(s)** and satisfaction of all requirements imposed by such execution.
user controls (1)	**Controls** created, modified, or selected by a user to control a particular **use** or **access** by the user to particular protected information.
validity	The state in which **authenticated** data conforms to predetermined completeness and consistency parameters.
virtual distribution environment	See Global Construction of **VDE.**